

## REMARKS

### **Pending Claims:**

In this application, claims 1-6 and 15-20 are currently pending. Although no amendments have been made to the claims, the claims have been reproduced above for the convenience of the Examiner.

### **Rejection under 35 U.S.C. §102: Claims 1-6**

The Examiner has rejected claims 1-6 as anticipated by Butler, U.S. Patent No. 6,917,943, while claims 15-20 were rejected as obvious over Butler in view of Bruckner, U.S. Patent No. 6,208,992. The office action considered Figure 2 of Butler to show the cells of the present invention. Bruckner is used to show the linking method of the present invention.

The applicant believes that the cited prior art in this fourth office reveals nothing significantly different than the prior art that has been cited in office actions one and three. In the applicant's mind, the Examiner has considered and accepted the applicant's arguments on these points twice before. Consequently, with the Examiner's indulgence, the applicant would like to review the present invention and the prior office actions in order to aid the Examiner in understanding these similarities.

### ***The Present Invention: Data Cells (Claims 1-6)***

Claim 1 covers a collection of data comprising “a plurality of data cells,” with each cell being a data construct. Like other data constructs used for the storage of data, the cell data construct of claim 1 contains an attribute value associated with an attribute type. Unlike other data constructs, each cell of newly amended claim 1 must contain “the attribute value for *only the single attribute type* and for *only the specific instance of the specific entity type*.” To further clarify this, claim 1 specifically states that the cell “does not contain the attribute value for any other attribute type or any other instance of the specific entity type.” Thus, in order for a reference to anticipate claim 1, the reference must teach a data construct that contains the attribute value for only the single attribute type and for only the specific instance of the specific entity type and no other attribute value.

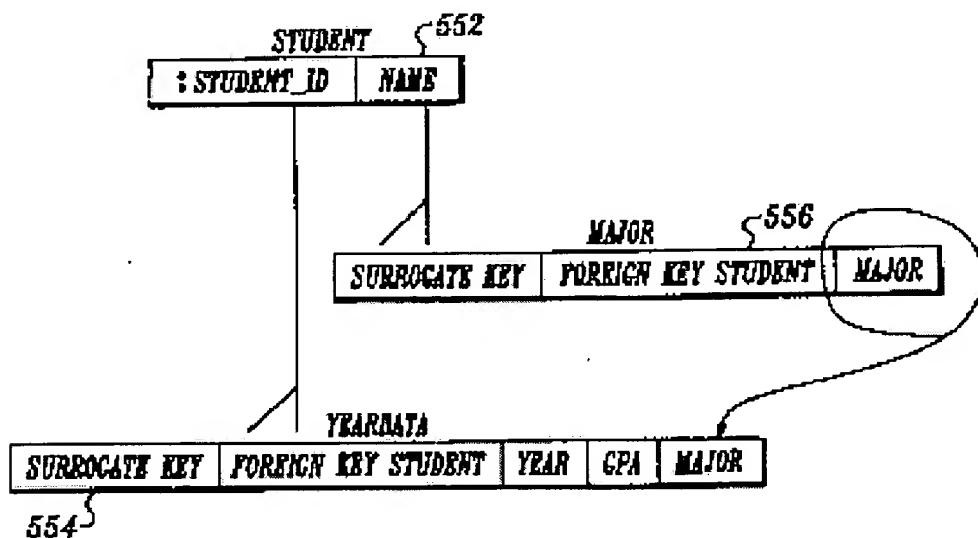
In addition to the attribute value, claim 1 requires that the cell data structure contain within itself a single instance identifier and a single attribute type identifier. Claim 2 adds that the cell must also contain an entity identifier value. It is these four elements (instance identifier, attribute type identifier, attribute value, and entity identifier) that allow each cell to be "self-identifying," which is an additional limitation of newly amended claim 1. The self-identifying nature of each cell is vital for the functioning of a data collection made up of data cells, as is explained in the Specification:

Because there is no external construct to associate one cell 100 with another, each data cell 100 of the present invention must be self-identifying. In other words, the data cell 100 must contain not only the value of interest, but it also must contain enough information to identify the attribute to which the value relates, and to associate the attribute with a particular instance of an entity.

Specification, page 7, lines 12-19.

***Office Action Number One: July 23, 2003***

This first office action rejected claims relating to the data cells as being anticipated by Kawai, U.S. Patent No. 5,717,924. The office action stated that Kawai describes a collection of data cells as shown in Figure 12a. Figure 12a from Kawai is set forth below.



***FIG. 12A.***

The applicant responded to this rejection with the five major arguments that are enumerated below and which are accompanied by text taken from the applicant's Response:

1. That the office action is ambiguous as to which elements in Kawai teaches the data cell data construct as defined in claim one:

The office action asserts that Figure 12a of Kawai teaches a plurality of data cells. As explained above, Figure 12a is seen to show four different data constructs: semantic data object 550, and the three relational tables 552, 554, and 556 that implement object 550. The office action is ambiguous as to which of these standard database constructs is considered to be a data cell, and in fact implies that the figure as a whole shows a data cell. ... Thus, it is inappropriate to look at the four separate data constructs in Figure 12a as together defining a data cell, because this would be inconsistent with the language of amended claim 1.

2. That the relational database table in Kawai cannot be a data cell:

Looking at the individual elements of Figure 12a, it is clear that the tables and objects of Kawai are not data constructs that contain a single element of data. The student object 550, as is typical of all data objects, can store multiple components or attributes (such as the student id, major, and year-data). Similarly, the tables of Kawai, like all database tables, contain multiple rows with each row having multiple columns of data. Thus, neither the student object 550 or the relational tables 552-556 are data constructs that contain a single element of data, and therefore do not constitute a data cell as defined by claim 1.

3. That a single row in a relational database table is not a single data construct as required by the claims:

A single row in "Major" table 556 comes closest to being a data cell, since each row in table 556 contains only a single real data or attribute value column ("Major"), along with a surrogate key (similar to an object identifier) and a foreign key column. However, a row is not a separate data construct, but merely a portion of a table construct. This is clearly the case when one realizes that a row cannot stand on its own. It is relevant only within the definition of a table, which indicates the type of information that is found within each row through its column definitions.

4. That even if a single row in a relational database table were a single data construct as required by the claims, such a row does not meet the definition of a database cell as defined by the claims:

But even if one assumes that a row in table 556 is a separate data construct (an assumption that the applicant rejects), that row would not meet the other requirements of the claims. Specifically, there is no attribute type identifier (as required by part ii of claim 1) in that row, and there is no entity identifier value (as required by claim 2) in that row. The specification states that the attribute type identifier indicates the type of information found in the cell, such as an employee name. Specification, page 8, lines 8-9; page 3, lines 13-14. This is consistent with the limiting language of claim 1, wherein the attribute type identifier identifies "one specific attribute type for the specific entity type." The office action identifies the foreign key student of Figure 12a in Kawai as an attribute type identifier. The foreign

key student column contains the foreign key that relates these tables 554, 556 with the parent table 552. See col. 5, line 60 to col. 6, line 2. These columns do not identify an attribute type, and hence are not attribute type identifiers. Furthermore, claim 1 requires that the “attribute value” be for the attribute type specified in the attribute type identifier. Assuming the value in the “Major” column to be the attribute value, it is clear that this value is not for the attribute type identified in the foreign key student column.

Similarly, there is no entity identifier value in a row of table 556. The specification states the entity identifier value identifies the type of entity associated with a cell. Specification, page 7, lines 25-27. There is no such value in the rows of table 556. The office action cites “major fig. 12a, Kawai” as teaching this element. The applicant accepts that a “major” is a separate entity in the relational database constructs shown in Figure 12a. However, claim 2 does not simply require that an entity exist, but that each data cell contain an entity identifier value. As explained above, the entire of Figure 12a is not a cell data construct, but is in fact multiple constructs, namely an object construct and three table constructs. None of these constructs contain a single element of data, except a row in table 556, and that row does not have an entity identifier value.

5. That relational database tables of Kawai did not meet the specific requirements of the cell set that is claimed in claim three:

The office action then states that the limitation in claim 3, namely identifying a cell set as a group of cells having the same instance identifier value and the same entity identifier value, is taught in object 100 of Figure 2. This figure shows a simpler example of the ability of the Kawai invention to alter a relational database schema upon a change to a semantic object. In this figure, an employee object 100 is altered by adding a single-valued “Birthday” component. Kawai, col. 5, lines 5-42. This is accomplished by adding a “Birthday” column 107 to the existing relational table 105 representing an employee. Kawai, col. 5, lines 23-33. There is no teaching of data cells, instance identifier values, or entity identifier values in Figure 2 or its corresponding text. Furthermore, nothing in Figure 2 teaches or suggests defining a cell set as all cells having the same instance identifier value and the same entity identifier value.

#### *Office Action Number Two: March 4, 2004*

The Examiner responded by accepting the applicant's arguments that the relational database tables of Kawai did not teach a data cell as defined by claims 1-6. Consequently, in the second office action, the Examiner allowed claims 1-6 and 15-20, finding that the prior art did not anticipate or suggest the data cell concepts embodied in these claims. Rather than arguing the rejection of the remaining claims over newly cited art, the applicant elected to cancel these claims and let the allowed claims issue. Instead, however, the third office action was received.

*Office Action Number Three: December 6, 2004*

The third office action rejected claims 1-6 as anticipated by Berner, U.S. Patent No. 5,907,846, while claims 15-20 were rejected as obvious over Berner in view of Cheng, U.S. Patent No. 6,356,896. Cheng is used to cite the existence of one table sharing columns with two other tables. This office action considered Figure 3 of Berner to show the cells of the present invention:

302			
ID.	NAME	AGE	GENDER
111	JANE D.	22	FEMALE
222	JOHN D.	21	MALE
:	:	:	:
:	:	:	:
:	:	:	:

PERSONNEL DEPARTMENT VIEW

304			
NAME	ID	JOB GRADE	SALARY
JANE D.	111	L5	35000
JOHN D.	222	L4	28000
:	:	:	:
:	:	:	:
:	:	:	:

PAYROLL DEPARTMENT VIEW

306			
ID	NAME	PROJECT NAME	LOCATION
111	JANE D.	ALPHA	4TH FLR
222	JOHN D.	ALPHA	4TH FLR
:	:	:	:
:	:	:	:
:	:	:	:

*FIG. 3*

The applicant responded to this rejection in a manner very similar to its response to the first office action and the Kawai reference. The Applicant noted the similarity between the present office action and the office action of July 23, 2003, which cited the Kawai in place of Berner, and Williamson in place of Chang. The applicant then made the following arguments:

1. That it is unclear which element of Berner is being considered the data cell:

None of the cited prior references discloses or suggests a self-identifying data construct that contains the specific limitations of claim 1. Figure 3 of Berner teaches three logical views to the Employee and Project class shown in Figure 2, with each logical view being represented as a table with four fields. Berner, col. 6, line 53 to col. 7, line 21. Technically, these tables 302-306 are not separate data structures, but are merely "logical views" of the object classes shown in Figure 2. Berner, col. 7, lines 17-21. Nonetheless, the Applicant acknowledges that the creation of the tables shown in Figure 3 as actual data constructs is well within the scope of the prior art.

2. That the relational database tables of Berner cannot be the data cell of claim 1:

However, while these tables are data constructs that contain attribute values, these tables do not meet the specific limitations of claim 1. Specifically, the tables do not contain "only the one specific attribute type and for only the one specific instance of the specific entity type. The tables of Berner, like all database tables, contain multiple rows with each row containing attribute values for a different instance of an entity.

For instance, the first row in table 302 contains attribute values for Employee 111, while the second row contains attribute values for Employee 222. Inside of each row are multiple fields of data, with each field containing a separate attribute value (i.e., the ID, name, age, and gender of a specific employee). Thus, the tables 302-306 are no different than any prior art data table, and contain multiple attribute values for multiple instances of an entity. Thus, the tables cannot be considered a data construct containing an attribute value for only the single attribute type and for only the specific instance of the specific entity type. The clarification of claim 1 that the cell “does not contain the attribute value for any other attribute type or any other instance of the specific entity type” makes it even clearer that a database table such as those of Berner does not constitute a cell as defined by claim 1.

3. That a single row in the Berner tables is not a separate data construct:

Although not stated as such in the office action, one might consider whether an individual row in a data table (or even the content of an individual field inside of a row) might constitute a cell as defined by claim 1. Given the language of the amended claim 1, it is clear that this is not the case. First, a single row in a prior art table is not considered a separate data construct, but merely a portion of a table construct. This is clearly the case when one realizes that a row cannot stand on its own. It is relevant only within the definition of a table, which indicates the type of information that is found within each field through its column definitions. Second, a typical row in a database table will contain multiple attribute values for a single entity. This second distinction, however, may not apply to single column database table, in which each row has only a single attribute value. In this case, the row is effectively a single field value, such as “Jane D.” This then becomes mere data, not a data construct.

4. That even if a single row in the Berner relational database table were a single data construct, such a row does not meet the definition of a database cell as defined by the claims:

But even if one assumes an individual row or field in a table is a separate data construct, prior art database rows and fields do not meet the other requirements of the claims. Specifically, claim 1 requires that each cell have a single instance identifier, a single attribute identifier, and an attribute value. Claim 2 adds the requirement for an entity identifier value in that cell. To avoid having the definition of a cell in claims 1 and 2 cover any four element data construct, the claims have been carefully constructed to define these four elements as follows: [table removed].

A single field within a row (or a one-column table row) certainly does not have all of these elements. Furthermore, rows in a prior art database table, such as the rows in Berner’s tables 302-306, do not contain these elements. The office action states that the instance identifier is “ID 111,” the attribute type is “job grade K5 (L5),” the entity identifier value is “ID 111,” while the attribute value is not specifically identified. The office action also states that the specific entity type is “name Jane.” To the best of the Applicant’s understanding, the office action is mixing row values with field names. If one looks only at the values within a row of a table, which one must do if one considers the row itself to be the cell data construct, the office action has assigned “111” as the instance identifier value, “K5” as the attribute type, and either “111” or “Jane” as the entity identifier value. This is not consistent with the claim limitations found in claims 1 and 2, as the instance identifier must identify an instance of the entity type, which would mean that “111” defines an instance of type “Jane.” This is

clearly not correct. In addition, the attribute value must be a value for an attribute type, meaning that the "35000" value in the row is a value for attribute type "K5," which is also inaccurate. There is no variation in how the fields in the database tables of Berner are mapped to the claim elements of claims 1 and 2 that will allow the Berner table to anticipate claim 1. The attribute type identifier must appear within the same data construct as the attribute value. The entity type identifier must also appear with the same data construct as the instance identifier. If a single field or row of a table is compared to this definition of a cell, it will not have these elements. If the entire table is compared to this definition, the table will be found not to meet the requirement that each cell contain the attribute value for only the specific attribute type and only the specific instance of the specific entity type.

5. That the relational database tables of Berner did not meet the specific requirements of the cell set in claim three:

The office action also states that the limitation in claim 3, namely identifying a cell set as a group of cells having the same instance identifier value and the same entity identifier value, is shown by the fact that each Berner table 302, 304, and 306 contain a row with a name value of Jane and an ID value of 111. If this were to meet the limitation of claim 3, then one of these two values must be the entity identifier identifying the specific entity type. Assuming that 111 is the instance identifier, the value Jane would have to identify the type of entity for the cells (meaning that this cell relates to instance 111 of all Jane-type entities). However, Jane is clearly an attribute value for a particular attribute type (name), and the entity type is an Employee or Person. Under the language of the claims, the entity identifier must identify the entity type and must be repeated in the cells along with the instance identifier in order to define a cell set. This is not shown in Berner.

6. And finally, that the database elements of Berner are not self-identifying as was then required by the claims:

Furthermore, a standard row in a table will not meet the claim requirement that each cell be self-identifying. In a prior art databases, data identifying information is built into the construct of a table or the construct of the object that contains the data. Outside of the context, the data itself means nothing:

111	Jane D.	L5	35000
-----	---------	----	-------

This is just random data stuck together, with no indication as to the meaning of the numbers "111" or "35000." When this information is put into the context of a predefined table, however, the data has value:

ID	Name	Job Grade	Salary
111	Jane D.	L5	35000

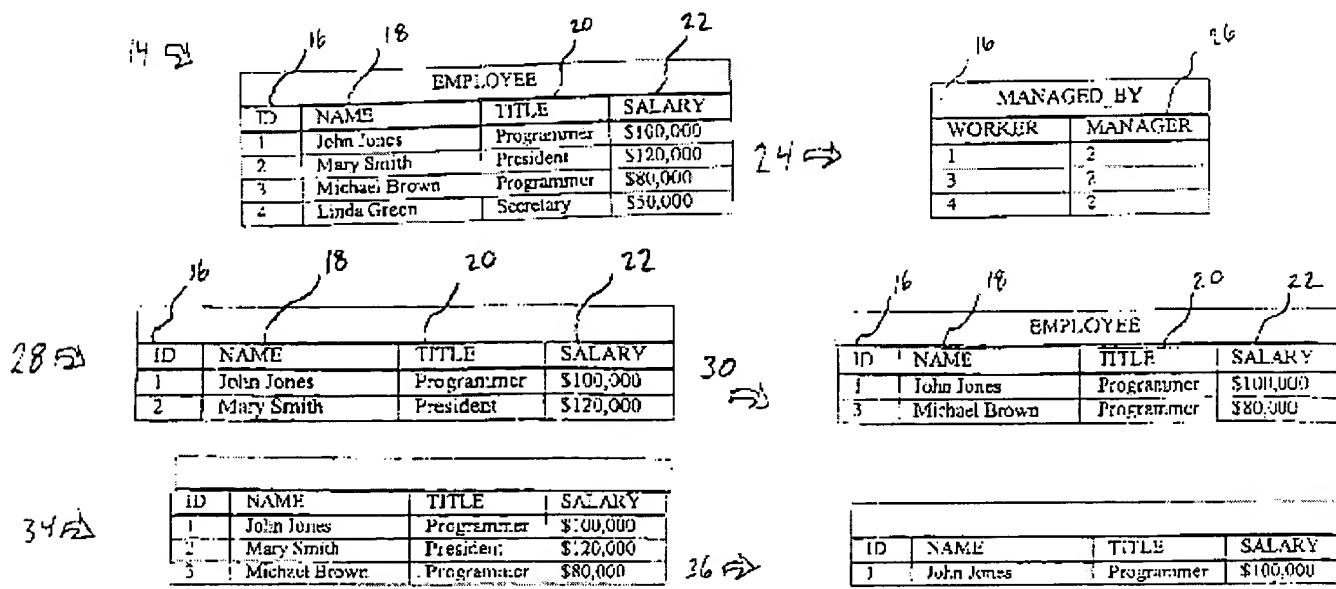
In contrast, when this information is put into the self-defining data cells of the present invention, it exists as pure data outside of any external construct and yet maintains its value:

Employee	111	Name	Jane D.
Employee	111	Job Grade	L5
Employee	111	Salary	35000

This is because every data cell defines not only an attribute value ("Jane D."), but also the attribute type (Name) associated with that attribute value. Similarly, the cell of the present invention defines not only the instance identifier (111), but also the entity of which this cell is an instance (Employee). Once a cell data construct contains not only the attribute value, but the attribute type and the instance identifier, it becomes self-identifying. This self-identifying nature of each cell is an intricate part of the present invention. This has been added as an element of claim 1, and is not taught or suggested in the prior art.

#### *Current Office Action: July 28, 2005*

The Examiner has again accepted the applicants arguments that the relational database tables shown by Berner do not teach a data cell as defined by claims 1-6. However, the current office action now rejects these claims over Butler, stating that Figures 2A, 2B, 2C, 2D, 2F, and 2G show the database cell of claims 1-6. These Figures are reproduced below:



The applicant respectfully submits that these relational database tables teach nothing beyond that which was taught in the Kawai and Berner. In claiming that these tables show the data cells of claims 1-6, the current office action does not respond to any of the arguments successfully presented by the applicant in response to the rejections

over Kawai and Berner. The arguments used previously to successfully distinguish Kawai and Berner apply equally to Butler, and therefore are repeated:

1. **There is no identification of which element is the data cell.** The office action does not specifically point out which table or which portion of the table constitutes the data construct that comprises a data cell of claim one.
2. **No prior art relational database table can be a data cell.** This is because the tables do not meet the claim limitation that the cells contain "only the one specific attribute type and for only the one specific instance of the specific entity type" Furthermore, the tables cannot meet the limitation in claim 1 that the cell "does not contain the attribute value for any other attribute type or any other instance of the specific entity type." The tables of Butler are no different than the tables of Kawai or Berner, and contain multiple attribute values for multiple instances of an entity.
3. **A single row in a relational database table is not a data construct with a single attribute value.** A single row in a prior art table is not considered a separate data construct, but merely a portion of a table construct. In addition, a typical row in a database table will contain multiple attribute values for a single entity.
4. **A single row in a relational database table does not meet the definition of a data cell as defined by claims 1-6.** Claim 1 requires that each cell have a single instance identifier, a single attribute identifier, and an attribute value. Claim 2 adds the requirement for an entity identifier\_value in that cell. These four elements are defined as follows:

a single instance identifier value	identifying one specific instance of a specific entity type
a single attribute type identifier value	identifying one specific attribute type for the specific entity type
an attribute value	for the identified one specific attribute type
a single entity identifier value	identifying the specific entity type

A single field within a row does not have all of these elements, nor does any table row shown in any of the prior art references. The current office action labels column 16 (the ID column) as the single instance identifier, column 18 (name) as the specific entity type, column 22 (salary) as the attribute type, and "salary" as the attribute value. But this assignment of

values does not work. If we were to examine a single row of table 14, the first row does not contain instance "1" of type "John Lucas." There are not separate instances of John Lucas, each with a different salary.

Furthermore, the office action assigns the salary to both the attribute value and the attribute type. Consequently, as explained previously in connection with Kawai and Berner, no single row in Butler contains these four required elements. This is because the data in each row has the benefit of the table definition from which each of the fields in the row can draw its meaning. Information about the entity type and attribute type are built into the structure of the table itself (in the column name and definition), and therefore are not found within the table row itself.

5. **The prior art relational database table does not show a set of cells where all cells have the same instance identifier and same entity identifier.** The office action refers to column 3, lines 60-67 of Butler for this teaching. However, this section merely describes the intersection of two select commands ("select salary >= 100000" and "select title = Programmer"). There is no set of multiple cells, nor is there any attempt to define a set that contains "all of the data in the collection relating to the one specific instance of the one specific entity type." This requirement of claim 3 is simply not found in the cited section of Butler.
6. **Nothing in the prior art of relational databases teaches the self-identifying aspect of a data cell.** The argument from the applicant's response to the third office action will be redrafted with the data found in Butler in order to further explain this advantage of the present invention:

When a cell data construct contains not only the attribute value, but the attribute type and the instance identifier, it becomes self-identifying. This self-identifying nature of each cell is an intricate part of the present invention. In contrast, Butler requires the relational table definition for its data to be identifiable. A single row in Butler does not identify itself:

1	John James	Programmer	\$100,000
---	------------	------------	-----------

This pure data gives no indication as to what this data relates. Is "John James" the name of a company to which a programmer is assigned? Is the value "1" indicative of the company's rank in the Fortune 500? Only when the entire table is defined can we identify the significance of the data:

EMPLOYEE			
ID	NAME	TITLE	SALARY
1	John James	Programmer	\$100,000

Now can we identify this as a list of employees, that "1" is the employee ID and "John James" is the name of the employee. In contrast, when this information is put into the self-defining data cells of the present invention, it is self-identifying:

EMPLOYEE	1	NAME	John James
EMPLOYEE	1	TITLE	Programmer
EMPLOYEE	1	SALARY	\$100,000

No information is needed from a table structure in order to give meaning to this data. Thus, while the data cells in the present invention are self-identifying, relational database tables such as those shown in Butler, Berner, and Kawai are not.

#### Rejection under 35 U.S.C. §103: Claims 15-20

Claim 15 defines a collection of data cells, with a first and second data cell each having the same format and fields. To anticipate claim 15, a prior art reference must teach a linking cell "defining an association between the first cell and the second cell" by combining the same two fields from the first and second cell into the four fields of the linking cell.

These elements are not found in the prior art. The cited section of Bruckner teaches ordinary prior art link tables with a key from both a first and second table. Bruckner does not start with a link between two identically formatted cells with the same format and fields. Claim 15 requires this element because it defines a method for linking identically formatted data cells. The combination of Bruckner with Butler does

not fill this gap, as Butler does not teach data cells having the same format and fields. Furthermore, while claim 15 requires the same two fields to be used from two identical cells to create a link, Bruckner only teaches the use of data from different key fields (i.e., the key fields in the individuals table and the cars table) to form an association. Even if an individual is linked to an individual, only a single key field from each record (not the two fields required in claim 15) is used to make the association.

Claim 16 adds an additional requirement that the linking cell also have the same format as the two data cells being linked. This vital claim element allows the linking cell to be joined seamlessly into the entire cell collection defined in claim 15. Only by having data cells and linking cells formatted identically can certain efficiencies in the present invention be achieved. Nothing in Bruckner nor Butler teaches or suggests this aspect of the present invention. The office action cites Figure 2 of Bruckner for this teaching. However, Figure 2 shows a table that contains multiple objects, with each object being represented by a unique key, descriptor, and object type. Bruckner, col. 9, lines 9-24. In contrast, relationships in Bruckner are stored in a separate table of a different structure, specifically the table of Figure 4. Bruckner, col. 10, lines 12-34. Thus, relationships and objects in Bruckner are stored in separate, differently formatted tables.

Claim 17 further includes a requirement that the linking cell utilize a flag to indicate that the cell contains linking information, a limitation that is not found in the cited prior art. The cited section in Bruckner does not discuss relationships, links, or flags, and therefore does not appear to be relevant to claim 17. Claim 18 defines the four fields of the cells as being an entity instance field, an entity type field, and attribute type field, and an attribute value field. This is not found in the cited prior art. While the office action cites Bruckner as teaching these elements, Bruckner shows only another example of a relational database table and therefore does not teach these elements (as is explained in more detail above in connection with claims 1 and 2). Claim 19 requires that all three cells contain an entity instance field, an entity type field, an attribute type field, and an attribute value field, and that particular fields of the two linked cells are found in specific fields of the linking cells. This is not taught in the cited section of Bruckner, which instead describes how previously defined relationships can be linked to attributes. Bruckner, col. 13, lines 39-62. This is not a description of how relationships between objects are created, which occurs using the table of Figure 4. Furthermore, the link described between relationship and attributes uses only a single key from a

relationship and a single key from an attribute. Nothing in this description teaches or suggests the creation of a linking cell in the way claimed in claim 19. Finally, claim 20 depends from claim 19, and requires the creation of a second linking cell. The office action again cites Bruckner, col. 13, lines 39-62 for this teaching. However, there is no such teaching in Bruckner.

### CONCLUSION

All of the claims remaining in this application should now be seen to be in condition for allowance. The prompt issuance of a notice to that effect is solicited.

Respectfully submitted,  
INFOBIONICS  
By its attorneys:

Date: 1/26/06



Daniel A. Tysver  
Registration No. 35,726  
Beck & Tysver, P.L.L.C.  
2900 Thomas Ave., #100  
Minneapolis, MN 55416  
Telephone: (612) 915-9634  
Fax: (612) 915-9637